LICEUL TEORETIC “ONISIFOR GHIBU”

SIBIU

PROIECT PENTRU ATESTARE

PROFESIONALĂ

an școlar: 2024-2025

COMPETENȚE CERTIFICATE:

I. Realizarea design-ului și structurii produselor

soft necesare implementării de: sisteme

software, aplicații software, baze de date,

pagini WEB. (software orientat client)

II. Particularizarea, configurarea și modificarea

aplicațiilor software, în scopul adaptării

sistemelor informaționale ale clientului.

ELEV: Monoran David

CLASA: a XII-a I

PROFESOR COORDONATOR : Avram Monica

TEMA PROIECTULUI:

Sistem de prezenta pe baza cardului RFID în C# si arduino.

CUPRINS

Motivarea alegerii temei ....................................................................................

Aspecte teoretice ................................................................................................

Software utilizat …………………………………………………………

Algoritmi utilizați/noțiuni teoretice utilizate ……………………………

Prezentarea aplicaţiei............................................................................................

Concluzii .................................................................................................................

Bibliografie/Webografie …………………………………………………………

# Motivarea alegeri temei

Am ales aceasta tema deoarece are aplicatii practice nenumarate, putand fi introdusa in companii ca un mod de monitorizare a prezentei angajatiilor sau a salilor accesate de acestia, la a fi folosit de cluburi sportive de Yoga sau arte, pana la contorizarea prezentei studentiilor la cursuri sau laboratoare.

De asemenea,

Aspecte teoretice

Proiectul de atestat este format din 3 componente:

* Web Api-ul a fost realizat in C# in mediul de programare „Visual Studio 2022” si acceseaza un server de baze de date SQL pentru a stoca informatii precum detaliile utilizatoriilor, prezenta acestora si altele.
* ??Web App-ul a fost realizat cu Razor Pages, care imbina structura pagini scrisa in HTML si scirpt-urile aferente scrise in C#.
* Hardware-ul, care consta dintr-un LCD, cititor RFID si un microcontroller ESP32 programat in C++ in ArduinoIDE.

Software utilizat

* „Visual Studio 2022” a fost folosit pentru programarea WebApp-ului si al WebApi-ului
* „Postman” a fost folosit pentru testarea metodelor implementate de WebApi.
* „Github” a fost folosit pentru source control.
* „SQL Server Management Studio” a fost folosit pentru conectarea la serversul de baze de date si verificare datelor introduse de WebApi.
* „ArduinoIDE” a fost folosit pentru programarea microcontrollerului ESP32.
* „Fusion360” a fost folosit pentru realizarea carcasei si a schematicii.

Prezenatre software

* Visual Studio 2022 este un mediu de dezvoltare integrat (IDE) care este foarte popular pentru sarcinile de lucru .NET și C++. Această ofera unelte valuroare pentru crearea de cod precum:
  + Depanarea (debugging) – instrumente puternice precum Live Debugging, Watch, Breakpoints avansați și IntelliTrace ajuta la indentificarea si rezolvarea erorilor.
  + Completarea codului – IntelliSense și IntelliCode oferă sugestii inteligente, personalizate în funcție de stilul de programare si restul cod-ului.
  + Ecosistemul vast de extensii inseamna ca este alegerea ideală pentru dezvoltatorii care lucrează la proiecte complexe, fie ele desktop, web, mobile sau cloud.
* Postman este un instrument care este foarte popular pentru sarcinile de lucru legate de testarea și dezvoltarea API-urilor. Postman oferă o interfață prietenoasă pentru a construi, trimite și vizualiza cereri HTTP precum GET, POST, PUT sau DELETE.
* GitHub este o platformă de găzduire a codului sursă care este foarte populară pentru sarcinile de lucru legate de controlul versiunilor și colaborarea între dezvoltatori.
  + GitHub oferă o interfață web ușor de folosit pentru a explora fișierele dintr-un proiect, a vizualiza istoricul modificărilor și a urmări cine a contribuit la ce parte din cod.
  + Completarea codului și integrarea – GitHub se integrează perfect cu editoare precum Visual Studio Code, oferind sugestii inteligente și acces direct la repository-uri.indiferent de locație.
* SQL Server Management Studio (SSMS) este foarte popular pentru sarcinile de lucru legate de gestionarea și dezvoltarea bazelor de date SQL Server. SSMS oferă o interfață grafică intuitivă care permite explorarea și administrarea tabelelor, procedurilor stocate, funcțiilor, și altor obiecte dintr-o bază de date SQL Server.
* ArduinoIDE este un mediu de dezvoltare integrat (IDE) care este foarte popular pentru sarcinile de lucru legate de programarea microcontrolerelor și dezvoltarea proiectelor cu Arduino.
* Fusion360 este un software CAM, CAD si CAE care este foarte popular pentru sarcinile de lucru legate de modelarea 3D, proiectarea mecanică, simulare și fabricație.

Prezentarea aplicatiei

WebApp

WebApp-ul reprezinta interfata grafica (front-end-ul) si codul legata de aceasta.

La accesarea initiala, utilizatorul va fi intampinat cu pagina de „Welcome” unde pot fi afisate informatii generale.

Acesta are posibilitatea sa se logeze pentru a avea acces la mai multe informatii in functie de ce rol are (Admin sau client).

La apasarea butonului de Login, utilizatorul va fi redirectionat pe aceasta pagina unde isi va introduce credentialele daca are cont deja existent.

Dupa ce utilizatorul s-a logat, acestuia ii se va arata una din 2 layout-uri, in functie de rolul acestia, putand fi client sau admin.

In interfata admin-ului se observa va exista mai multe optiuni, destinate controlului manual al datelor din baza de date, pentru toti utilizatorii, putand modifica orice.

In interfata clientului, acesta are acces la prezenta acestuia si la cursurile la care este inscris, putand doar vizualiza acestea fara sa le poata modifica. El isi poate modifica doar parola din pagina cu profilul.

Asa arata pagina cu profilul utilizatorului, continand username-ul, prenume, nume, institutia, email-ul si numarul de telefon.

Utilizatorul cu rol de client nu va acces la link-ul de „Edit”.

Utilizatorul isi poate schimba parola prin apasarea linkului „Change Password” . Acesta va fi nevoit sa-si introduca parola actuala si de 2 ori parola noua.

WebApi

WebApi-ul gestioneaza partea de back-end a proiectului, premitand accesul la baza de date. Butoanele de pe WebApp cat si unele pagini apeleaza metode din WebApi pentru a primi informatii din baza de date.

Metodele din WebApi sunt de tip HTTP, astfel trebuie selectat ce tip sa fie. In realizarea proiectului am folosit metodele: GET (obtinere de informatii), POST(creare intrari noi in baza de date), PUT(modificarea intrari din baza de date) si DELET(stergere din baza de date).

In structura acestuia sunt definiti 4 „utilizatori”, respectiv cei umani: clientul si admin-ul, respectiv scanner-ul, adica microcontroller-ul ESP32 si unul comun.

* „AdminController” este responsabil pentru metodele folosite de WebApp cand este logat utilizatorul cu rol de admin.
* „ClientController” este responsabil pentru metodele folosite de WebApp cand este logat utilizatorul cu rol de client.
* „CommonController” este resonsabil pentru metodele folosite de WebApp atunci cand nu este inca logat utilizatorul si contine metode precum resetarea de parola sau logarea.
* „ESP32Controller” contine o singura metoda folosita de microcontroller pentru a introduce in tabelul „Entries” prezenta clientilor.

Baza de date

In crearea si utilizarea bazei de date am optat pentru code first aproach, adica structura bazei de date este definita in totalitate de cod.

In prima parte sunt definite tabelurile, structura acestora continand pe post de coloane variabilele din interiorul claselor date.

In a doua parte se stabilesc cheile primare si ce campuri sa fie unice. De asemenea de aici se pot adauga date initiale, care vor fi adaugate o data cu crearea tabelelor.

Inserarea de date

Pentru a introduce si extrage din baza de date, am creat cate o clasa diferita pentru fiecare tabel, in care am injectat „AttendanceContext” pentru a avea acces la baza de date si am definit metodele pe care le va folosi Controllerele

Clasa mosteneste o interfata. Aceasta defineste ce functii trebuie sa existe si ce parametri trebuie sa aiba.

„AttendanceContext” a fost injectat pentru ca acesta trebuie devinit ca fiind „Singleton”, adica sa existe o instanta unica pentru tot WebApi-ul, din cauza modului in care EntiyFramwork face legatura cu baza de date.

Apelarea functiilor de inserare

La apasarea unui buton sau incarcarea unei pagini, WebApp-ul trimite un HTTP request catre WebApi. Acest request ii este asociat unei functii care va apela metodele pentru inserarea sau de extragere din baza de date.

Injectarea interfetelor se face tot din cauza modului de functionare a lui Entity Framework, ele fiind initializate ca fiine Singleton.

„Route” defineste url care ii este asociat functiei. De exemplu functia „GetAllUsers” are url-ul „https://localhost:7172/Admin/cleients” si este unul de tip GET.

Setari globale

Clasele si interfetele si initializarea lor de tip „Singleton”, si definirea locatiei bazei de date se realizeaza in „Program.cs”

„LaptopAttendanceDbConnectionString”, „PCAttendanceDbConnectionString” si UID-ul sunt definite in appsettings.json. Am decis sa se salvez aici pentru a evita harcodarea lor, care nu este recomandata.

Securitate

Parola utilizatoriilor nu trebuie salvata in forma simpla, deoarece reprezinta un risc major in cazul unei spargeri de baza de data. Din acest motiv, parola trebuie incriptata inainte de a fi sotcata. Parolei simple ii se adauga un „salt” care poate sa fie de exemplu data la care se incripteaza petru a face si mai dificila spargerea, iar apoi este trimisa unui algoritm special care returneaza parola criptata.

Din parola „testpass1”, aceasta devine „AQAAAAIAAYagAAAAEKedSrhwVrJx/bhw7PqwY3V9zfw7pR8/5SbMqDnRCW+hq9Vrs7/xybimErcAe9u+Zw==”. Daca recriptam din nou parola normala, o sa fie returnat o alta parola criptata.

Pentru compararea parolelor (de exemplu petru logare), „AspNetCore.Identity” care este responsabila pentru cripatrea parolei are o functie de comparare care returneaza >0 daca parolele sunt identice.

In fiecare metoda WebApi se compara un UID care vine impreuna cu requestul. Acesta are scopul de a face imposibila inserarea sau stergerea de date din baza de date de catre cineva/ceva neautorizat. Acesta poate fi asociat cu o parola necesara la fiecare comunicare.

Cerințe formatare:

-utilizare diacritice

-font Times New Roman de 14pt

-spațiere la 1,5 rânduri

-titlu de capitol 20pt, centrat

-subtitlu capitolului de 18pt

-margini stg 2,5cm, restul 2cm

-aliniere stânga-dreapta

-numerotare pagini-jos, centru (coperta, adică prima pagină, nu se numerotează)